USING THE FAST MARCHING METHOD FOR TIME ADVANCEMENT

David Salac and David Chopp
Department of Engineering Sciences and Applied Mathematics,
Northwestern University, 2145 Sheridan Road, Evanston, IL 60208

INTRODUCTION

The modeling of moving interfaces is both a challenge and necessity of computational materials science. Here we are interested in the front-capturing technique of level sets. Front capturing treats a moving interface implicitly through the use of some external mathematical construct. The advantages of front capturing schemes are ease of implementation and the natural way the methods handle large topological changes. Two prototypical examples of front capturing are the volume-of-fluid and level set methods. A disadvantage of front capturing is the additional computational cost associated with using the external mathematical construct. But this limitation can be overcome using advanced techniques such the method described below.

Here we introduce a novel time-stepping scheme for level set methods based on the fast marching method. The level set method captures moving interfaces by embedding the boundary into a function of one higher dimension (a two-dimensional curve is modeled using a three-dimensional function). By setting the interface to correspond to a specific level set of the embedding function (typically chosen to be the zero level set), we can implicitly track the interface through time. To advance forward in time a time-stepping scheme is required. Currently there are two main choices. The first is explicit schemes, which are computationally quick per time step but are subject to strict time-step size. The second are semi-implicit schemes which allow for larger time steps than explicit schemes but require the solving of large non-linear systems of equations which may be computationally expensive.

Instead we propose to utilize the fast marching method as our time-stepping scheme. Originally introduced by Sethian (1), the fast marching method was developed as an optimally efficient algorithm for solving problems of front evolution where the interface velocity is monotonic. Due to this limitation the fast marching method has only been utilized for reinitialization and velocity extensions. To our knowledge the use of the fast marching method for time advancement has not yet been achieved. The method proposed here is capable of modeling interfaces with a velocity sign change. The advantage of our method is that we can achieve better stability than explicit schemes at a cost much lower than a semi-implicit scheme. The method will be demonstrated by modeling typical test cases based on constant, position-dependent and curvature-dependent velocities.

FAST MARCHING METHOD FOR TIME ADVANCEMENT

Details of the algorithm will be available in (2). Here we present a short overview of the numerical algorithm. Begin by considering an arbitrary interface described using the level set method in a domain with a known set of computational grid points. We prescribe that the level set is a signed distance function, with the understanding that the negative level set region indicates the region of the computational domain occupied by the “body”, the positive level set region indicates the region in the computational domain not enclosed by the body and the zero level set corresponds to the interface of interest. We also prescribe a general normal velocity on the interface. We do not place any restrictions on the velocity, thus allowing for bi-directional. To advance forward in time the following general steps are taken. Please note that the time-of-crossing has the following meaning; given a point in the computational domain and assuming the velocity on the interface does not change, the time-of-crossing is the amount of time needed for the interface to reach the given point.

<u>The Scheme</u>

1. Calculate an initial time-of-crossing map for those grid points directly next to the interface.
2. Split the interface into two distinct and possibly open curves: the positive velocity interface and the negative velocity interface.
3. Use the Piecewise Fast Marching Method (3) to obtain two time-of-crossing maps, one for the positive velocity interface and one for the negative velocity interface.
4. Merge the two interfaces into a single time-of-crossing map.
5. Advance forward in time by setting the interface at the next time step to the level contour equal to the time step of the combined time-of-crossing map.

RESULTS

Preliminary results have shown that the scheme described above is capable of modeling bi-directional interfacial motion. We are currently testing numerical efficiency of the scheme, but experience has shown that the method is extremely computationally efficient. An example of motion of a shape under curvature flow is shown in Figure 1.

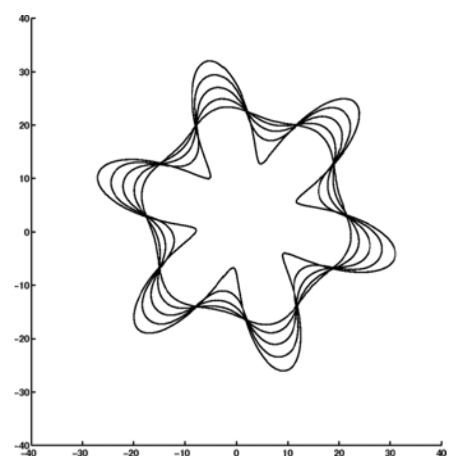

Figure 1. Motion of a star-shaped body under curvature flow.

ACKNOWLEDGMENTS

This work was supported in part by a grant from the NSF RTG program.

REFERENCES

1. J. Sethian, p. 1591-1595, Proceedings of the National Academy of Sciences 93 (1996).
2. D. Salac and D. Chopp. In preparation.
3. D. Chopp. SIAM J. Sci. Comput. Accepted